Dimensionality Reduction in Machine Learning: Taming High-Dimensional Data

Ahmed BADI
ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

January 2026

Abstract

Dimensionality reduction is a core tool in machine learning and data analysis. Many modern datasets contain dozens, hundreds, or even thousands of features. Working directly in such high-dimensional spaces can lead to slow computation, overfitting, and difficulties in visualization. Dimensionality reduction methods address this by transforming data into a lower-dimensional space that preserves the most important structure. In this article, we explain the motivation behind dimensionality reduction, clarify its relationship to feature selection and feature extraction, and present key techniques including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and nonlinear methods such as t-SNE and UMAP. For each method, we give an intuitive explanation, the main mathematical ideas, and practical guidance on when to use it. We also discuss the advantages and limitations of dimensionality reduction in real-world machine learning workflows.

Keywords: Dimensionality Reduction, Curse of Dimensionality, PCA, LDA, t-SNE, UMAP, Manifold Learning, Visualization.

1 Introduction

In low dimensions (for example, 2D or 3D), we can often understand data through simple plots. However, many real-world datasets have high dimensionality: hundreds of features in tabular data, thousands of pixels in images, or tens of thousands of word counts in text. Training models directly on such data can be problematic:

- **Computational cost:** More features mean more parameters and longer training times.
- **Overfitting:** Models may memorize noise instead of learning general patterns.
- **Visualization:** Human intuition works poorly beyond 3D; patterns become hard to see.

Dimensionality reduction tackles these issues by mapping data from a high-dimensional space $\mathbb{R}^p$ to a lower-dimensional space $\mathbb{R}^m$ (with $m \ll p$) while trying to preserve important structure. [1], [2]

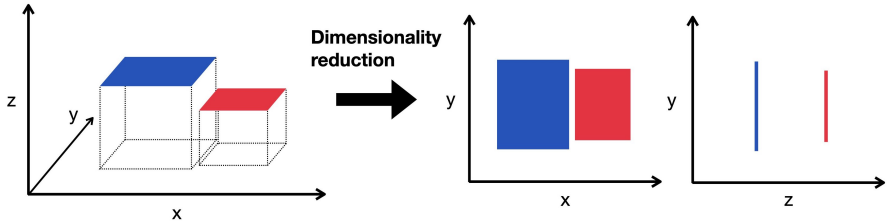

Figure 1: Illustrative example: projecting 3D data onto a 2D plane while keeping the main structure.

Dimensionality reduction is closely related to feature selection and feature extraction:

- Feature selection chooses a subset of existing features.
- Feature extraction creates new features as transformations of the original ones.

Dimensionality reduction usually refers to the broader process, often implemented via feature extraction methods like PCA, t-SNE, or UMAP. [3], [4]

2 Why Dimensionality Reduction Matters

2.1 Curse of Dimensionality

As dimensionality increases, data becomes sparse and distances behave strangely. Many algorithms (for example, KNN, clustering) rely on distance or density, and their performance can degrade in high dimensions. This phenomenon is called the *curse of dimensionality*. [5], [6]

Dimensionality reduction helps by concentrating the data into a lower-dimensional space where:

- Distances are more meaningful.
- Density estimation and neighborhood-based methods work better.

2.2 Benefits and Trade-offs

Benefits:

- **Faster computation:** Fewer dimensions often mean faster training and prediction.
- **Noise reduction:** Low-variance or noisy directions can be discarded.
- **Better visualization:** Techniques like PCA, t-SNE, and UMAP enable 2D/3D plots of complex data. [7]
- **Improved generalization:** By removing redundant dimensions, models may overfit less. [8]

Potential drawbacks:

- Some information is inevitably lost.
- Extracted features can be hard to interpret.
- Nonlinear methods can be computationally expensive and sensitive to hyperparameters.

3 Formal Definition

Let $\mathbf{x} \in \mathbb{R}^p$ be an original data point. Dimensionality reduction seeks a mapping

$$\phi : \mathbb{R}^p \to \mathbb{R}^m, \quad \mathbf{z} = \phi(\mathbf{x}),$$

with $m \ll p$, such that:

- $\mathbf{z}$ retains as much "useful information" as possible (variance, class separability, neighborhood structure, etc.).
- The mapping is usable for tasks like classification, clustering, or visualization.

For many methods, ϕ is linear:

$$\mathbf{z} = W^\top \mathbf{x},$$

where $W \in \mathbb{R}^{p \times m}$ is a projection matrix. For nonlinear methods, ϕ can be highly complex (for example, a neural network or a manifold embedding). [9]

4 Linear Dimensionality Reduction: PCA

Principal Component Analysis (PCA) is the most widely used linear dimensionality reduction technique. It finds orthogonal directions that capture maximal variance. [4], [10]

4.1 Covariance and Eigen-decomposition

Given centered data $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ and covariance matrix

$$C = \frac{1}{n} \sum_{i=1}^{n} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top,$$

PCA solves

$$C\mathbf{w}_k = \lambda_k \mathbf{w}_k,$$

with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. The eigenvectors $\mathbf{w}_k$ are principal directions, and the eigenvalues give the variance along each component.

The first m principal components for sample $\mathbf{x}_i$ are

$$\mathbf{z}_i = W^\top \tilde{\mathbf{x}}_i, \quad W = [\mathbf{w}_1, \ldots, \mathbf{w}_m].$$

4.2 Variance Explained

The fraction of variance explained by the first m components is

$$\text{ExplainedVariance}(m) = \frac{\sum_{k=1}^{m} \lambda_k}{\sum_{k=1}^{p} \lambda_k}.$$

In practice, we choose m such that this ratio exceeds a target (e.g., 90%).

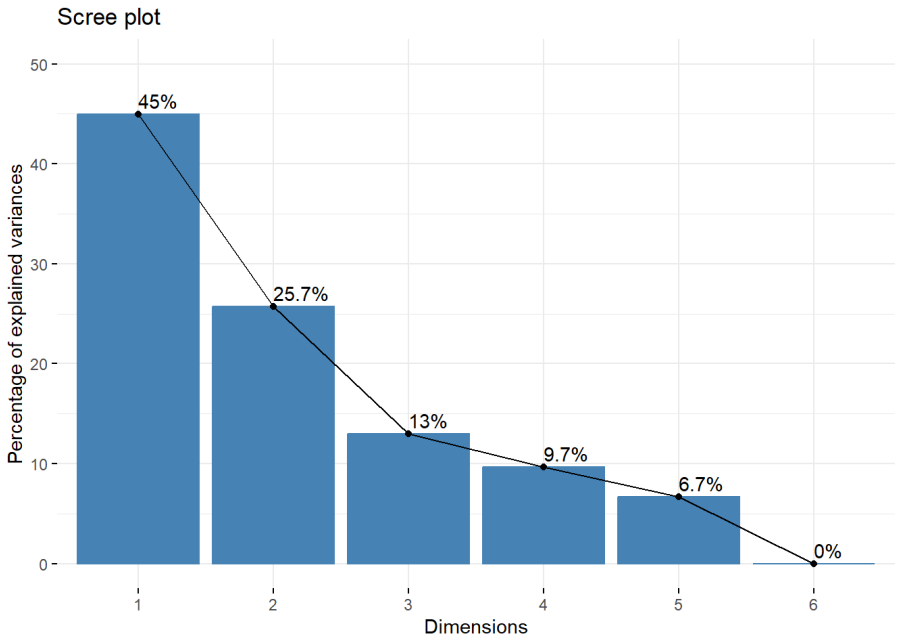

Figure 2: Scree plot of eigenvalues: choosing the number of principal components.

4.3 Advantages and Limitations

Advantages:

- Simple and computationally efficient.
- Provides decorrelated components.
- Useful for visualization and as a preprocessing step.

Limitations:

- Linear: cannot capture complex manifolds.
- Components are linear combinations, not directly interpretable features.
- Sensitive to scaling; standardization is often needed.

5 Supervised Linear Dimensionality Reduction: LDA

Linear Discriminant Analysis (LDA) can be used as a supervised dimensionality reduction technique that incorporates class labels. It aims to find projections that maximize class separation. [11]

5.1 Scatter Matrices

Define between-class scatter

$$S_B = \sum_{k=1}^{K} N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top,$$

and within-class scatter

$$S_W = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top,$$

where K is the number of classes, N_k number of samples in class k, $\boldsymbol{\mu}_k$ class mean and $\boldsymbol{\mu}$ global mean.

5.2 Optimization

LDA finds directions maximizing

$$J(\mathbf{w}) = \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}.$$

Solving

$$S_B \mathbf{w} = \lambda S_W \mathbf{w}$$

gives discriminant vectors. The data are then projected onto these directions to obtain low-dimensional features with good class separability. LDA yields at most $K - 1$ dimensions. [11]

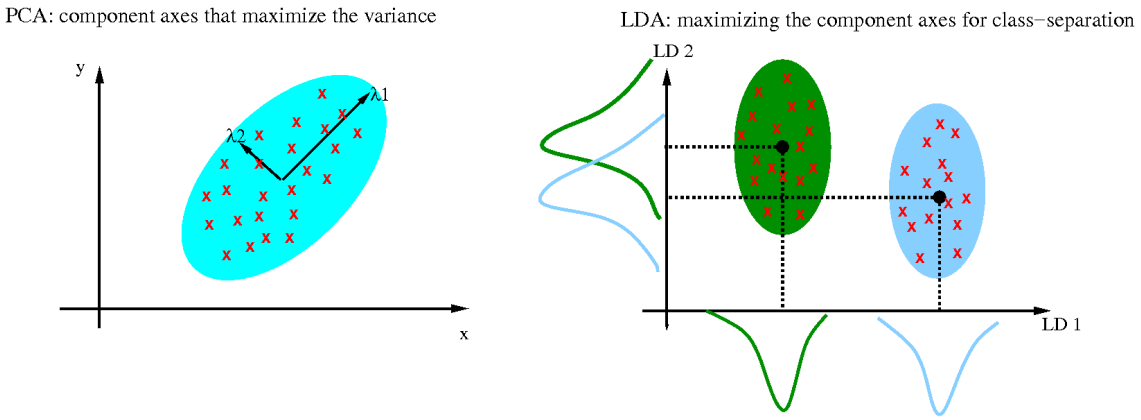

Figure 3: LDA projection: classes become more separable in the lower-dimensional space.

6 Nonlinear Dimensionality Reduction: t-SNE and UMAP

Many datasets lie on nonlinear manifolds embedded in high-dimensional spaces. Nonlinear dimensionality reduction methods aim to uncover these manifolds. Two popular tools for visualization are t-SNE and UMAP. [4], [6]

6.1 t-SNE: Preserving Local Neighborhoods

t-Distributed Stochastic Neighbor Embedding (t-SNE) builds probability distributions over pairwise distances in high and low dimensions. [12]

In high-dimensional space, it defines

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)},$$

and sets

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}.$$

In low-dimensional space, with points $\mathbf{y}_i \in \mathbb{R}^m$, t-SNE defines

$$q_{ij} = \frac{\left(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\right)^{-1}}{\sum_{k\neq \ell}\left(1 + \|\mathbf{y}_k - \mathbf{y}_\ell\|^2\right)^{-1}}.$$

It then minimizes the Kullback–Leibler divergence between these distributions:

$$\mathrm{KL}(P\|Q) = \sum_{i\neq j} p_{ij}\log\frac{p_{ij}}{q_{ij}}.$$

This objective encourages nearby points in high-dimensional space to stay close in the low-dimensional embedding, revealing clusters. [12]

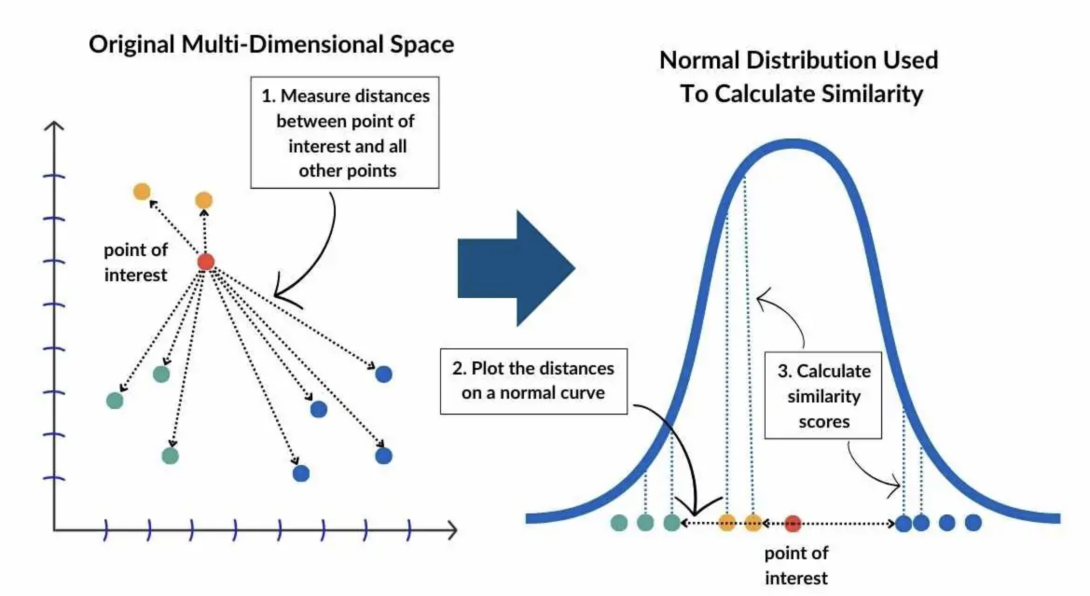

Figure 4: Example of t-SNE embedding of high-dimensional data into 2D clusters.

Source:
`https://i0.wp.com/spotintelligence.com/wp-content/uploads/2023/12/t-sne-visual-explanation.jpg`

6.2 UMAP: Manifold Learning with Graphs

UMAP (Uniform Manifold Approximation and Projection) is another nonlinear method that models data as a fuzzy topological structure (a graph) and optimizes a low-dimensional embedding that preserves this structure. [4], [13]

High-level steps:

- Construct a weighted k-nearest-neighbor graph capturing local relationships.
- Interpret this as a fuzzy simplicial set (a topological object).
- Optimize low-dimensional coordinates by minimizing a cross-entropy loss between high- and low-dimensional fuzzy sets using stochastic gradient descent.

UMAP often preserves both local and some global structure better than t-SNE and tends to be faster and more scalable. [14]

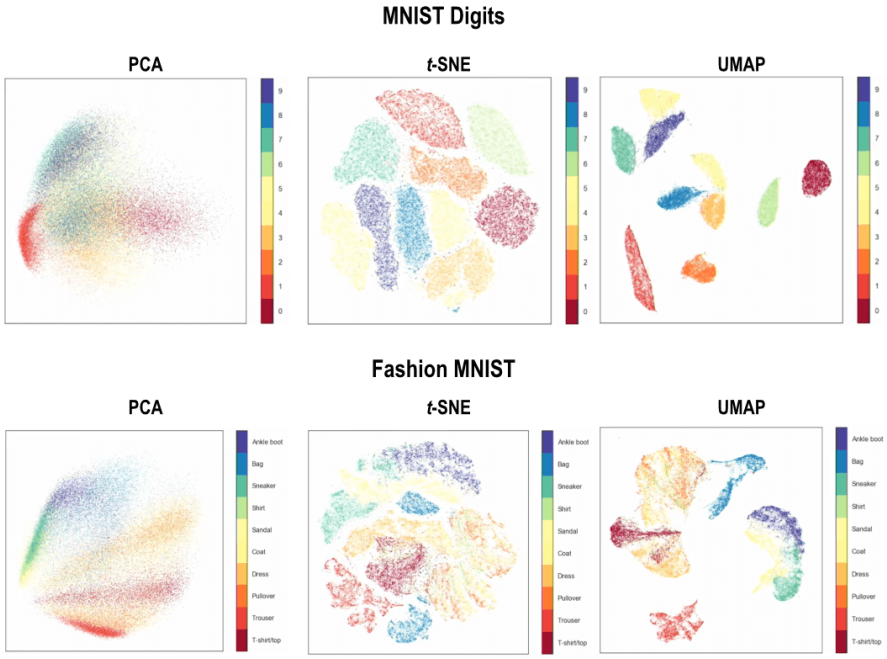

Figure 5: Comparison of 2D embeddings produced by PCA, t-SNE, and UMAP on the same dataset MNIST.

7 Linear vs Nonlinear Methods

The choice between linear and nonlinear dimensionality reduction depends on data structure and objectives. [8], [9]

Aspect	**Linear (PCA, LDA)**	**Nonlinear (t-SNE, UMAP, Autoencoders)**	**Typical Use**
Mapping	Linear projection	Complex nonlinear mapping	Visualization, embeddings
Structure	Captures global linear variance	Captures local manifolds and clusters	Complex data (images, text)
Interpretability	Higher	Lower	Depends on need
Compute cost	Lower	Higher	PCA for baseline; t-SNE/UMAP when needed

Table 1: Comparison of linear vs nonlinear dimensionality reduction methods.

A common practical strategy:

- Start with PCA: fast, provides a baseline.
- If patterns remain unclear, try t-SNE or UMAP for visualization.
- For complex tasks and lots of data, consider autoencoder-based dimensionality reduction.

8 Applications

Dimensionality reduction is widely used across domains. [2], [15]

- **Visualization:** Plot high-dimensional data (e.g., word embeddings, latent spaces) in 2D/3D.
- **Preprocessing:** Reduce dimensionality before clustering or classification to improve performance.
- **Noise filtering:** Remove low-variance components or noisy dimensions.
- **Compression:** Store lower-dimensional representations of images or signals.

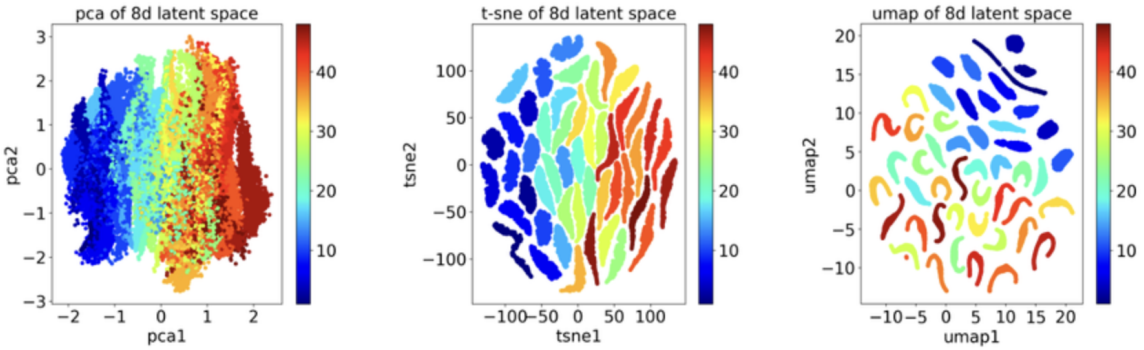

Figure 6: Latent space visualization of a model using PCA, t-SNE, and UMAP.

Source: ResearchGate — PCA, t-SNE and UMAP latent space

9 Practical Guidelines

Some practical tips when applying dimensionality reduction:

- **Scale features:** For PCA and many others, standardize features so each has mean 0 and variance 1.
- **Fit on training data only:** To avoid data leakage, learn the projection using only training data and apply it to validation/test sets.
- **Choose dimensionality carefully:** Use variance explained (PCA) or validation performance as a guide.
- **Use nonlinear methods mainly for visualization:** t-SNE and UMAP are excellent for exploring structure but less straightforward for downstream supervised modeling due to stochasticity and parameter sensitivity. [4], [7]

10 Conclusion

Dimensionality reduction is an essential tool for working with high-dimensional data. It helps:

- Mitigate the curse of dimensionality and improve model performance.
- Speed up training and reduce storage requirements.
- Reveal patterns and clusters through 2D/3D visualizations.

We have discussed:

- The motivation and formal definition of dimensionality reduction.
- Linear methods such as PCA and LDA.
- Nonlinear methods like t-SNE and UMAP for manifold learning and visualization.

There is no universally best method. PCA is often a strong baseline, while t-SNE and UMAP provide powerful tools for visual exploration of complex datasets. In practice, dimensionality reduction should be chosen and evaluated in the context of the downstream task, available compute, and interpretability requirements. [5], [8]

References

[1] GeeksforGeeks, *Introduction to dimensionality reduction*, https://www.geeksforgeeks.org/machine-learning/dimensionality-reduction/, 2025.

[2] Encord, *Top 12 dimensionality reduction techniques for machine learning*, https://encord.com/blog/dimentionality-reduction-techniques-machine-learning/, 2025.

[3] Statsig, *Feature selection vs. feature extraction: Key differences*, https://www.statsig.com/perspectives/feature-selection-vs-extraction, 2025.

[4] Aurigait Technologies, *Dimensionality reduction: Pca, tsne, umap*, https://aurigait.com/blog/blog-easy-explanation-of-dimensionality-reduction-and-techniques/, 2024.

[5] Y. Zhou et al., "A survey: Potential dimensionality reduction methods for data science," *arXiv preprint arXiv:2502.11036*, 2025.

[6] Wikipedia contributors, "Nonlinear dimensionality reduction," *Wikipedia*, 2024, https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction.

[7] LearnCodePro, *Dimensionality reduction: Visualizing data with pca, t-sne umap*, https://www.learncodepro.com/.../dimensionality-reduction-visualizing-data-with-pca-tsne-umap/, 2025.

[8] A. Kukushkin, "Manifold learning and dimensionality reduction: Algorithms, applications, pros, and cons," *Neptune.ai Blog*, 2025, https://neptune.ai/blog/dimensionality-reduction.

[9] APXML Course, *Linear vs non-linear dimensionality reduction*, https://apxml.com/courses/applied-autoencoders-feature-extraction/chapter-1-nn-dimensionality-reduction-recap/linear-vs-nonlinear-dimensionality-reduction, 2025.

[10] GeeksforGeeks, *Mathematical approach to pca*, https://www.geeksforgeeks.org/machine-learning/mathematical-approach-to-pca/, 2021.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[12] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[13] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[14] AI Competence, *Pca vs. t-sne vs. umap: Find the right fit for your data*, https://aicompetence.org/pca-vs-t-sne-vs-umap/, 2024.

[15] J. A. Lee and M. Verleysen, "Dimensionality reduction - an overview," *ScienceDirect Topics*, 2010.